

## PAJ

TI - LOGICAL SIMULATION DEVICE  
AB - PURPOSE: To simulate speedily a logic circuit of a large scale where the gate and function levels are mixed together.  
- CONSTITUTION: A data memory 26 is provided to store a node function together with a connection memory 25 which stores the connection information, an input memory 23 which stores the input value, an output memory 24 which stores the output value, an event memory 22 which stores the node whose input is changed, an index memory 21 which stores the address of the memory 23, a retrieving part 11 which retrieves the memory 21, an updating part 12 which updates those memories, an input synthesizing part 13 which synthesizes the node inputs, an arithmetic part 14, an event generating part 15 which detects the output change, and an event synthesizing part 16 which transfers the output to a connected destination. In such a constitution, the pipeline processing is carried out. Furthermore, an input control part 2 is added to issue the commands to each part together with an execution control part 3, an output control part 4, an input interface part 1 which performs the input/output events to the outside, and an output interface part 5.

PN - JP6004511 A 19940114  
PD - 1994-01-14  
ABD - 19940407  
ABV - 018200  
AP - JP19920165664 19920624  
GR - P1724  
PA - NEC CORP  
IN - HASHIMOTO KUNIHARU  
I - G06F15/20 ;G06F11/26

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平6-4511

(43) 公開日 平成6年(1994)1月14日

(51) IntCl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/20		D 7052-5L		
11/26	3 1 0	8323-5B		

審査請求 未請求 請求項の数2(全7頁)

(21) 出願番号 特願平4-165664

(22) 出願日 平成4年(1992)6月24日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 橋本 訓治

東京都港区芝五丁目7番1号日本電気株式会社内

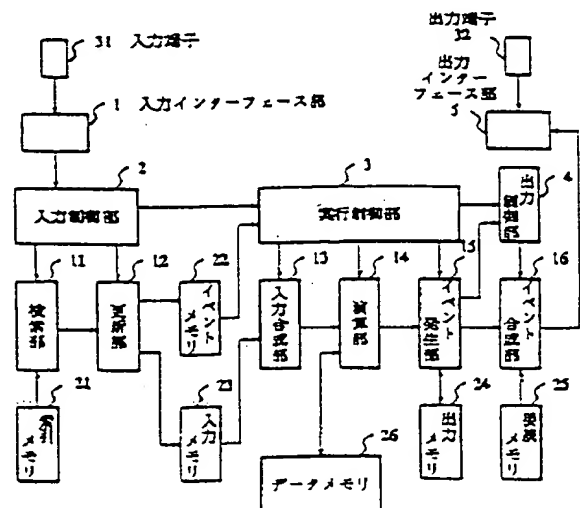
(74) 代理人 弁理士 京本 直樹 (外2名)

(54) 【発明の名称】 論理シミュレーション装置

(57) 【要約】

【目的】 ゲートレベルと機能レベルが混在した、大規模な論理回路のシミュレーションを高速に実行する。

【構成】 ノードの機能を記憶するデータメモリ26、接続情報を記憶する接続メモリ25、入力値を記憶する入力メモリ23、出力値を記憶する出力メモリ24、入力に変化したノードを記憶するイベントメモリ22、入力メモリのアドレスを記憶する索引メモリ21、索引メモリを検索する検索部11、メモリを更新する更新部12、ノードの入力を合成する入力合成部13、演算を行う演算部14、出力変化を検出するイベント発生部15、出力を接続先へ転送するイベント合成部16を設け、パイプライン処理を行う。また、各部に指令を発する入力制御部2、実行制御部3、出力制御部4、外部とイベントの入出力を行う入力インターフェース部1、出力インターフェース部5を有する。



## 【特許請求の範囲】

【請求項1】論理回路を、ゲートやメモリ等の機能を表すノードとノード間の接続情報とでモデル化し、ノードの出力変化をイベントとして次のノードに伝えて論理回路のシミュレーションを行う論理シミュレーション装置において、

ノードの入力情報を記憶する入力メモリと、ノードの出力情報を記憶する出力メモリと、ノードの接続情報を記憶する接続メモリと、ノードと入力メモリのアドレス値の関係を記憶する索引メモリと、イベントが伝播したノードの情報を記憶するイベントメモリと、その他のノード情報等を記憶するデータメモリと、

入力イベントを解析して検索部と更新部に指令を出す入力制御部と、イベントメモリを読み出してシミュレーションの実行指令を出す実行制御部と、発生したイベントの合成をイベント合成部に指令する出力制御部と、

入力イベントを格納する入力メモリのアドレスを索引メモリの情報から決定する検索部と、入力イベントを入力メモリの検索部から渡されるアドレス位置に、イベント情報をイベントメモリに記録する更新部と、入力メモリからシミュレーション対象ノードの入力イベントを読み出し、演算部に与える入力ベクタを合成する入力合成部と、入力合成部から与えられる入力ベクタとデータメモリからのノード情報を用いてノードの演算を行う演算部と、演算部から与えられるノードの演算結果を、出力メモリに保持されている以前のノードの出力と比較して、イベント発生の有無を決定し、かつ出力メモリを更新するイベント発生部と、接続メモリからノードの接続情報を読み出し、イベント発生部から受け取ったイベントを接続先のノードへ転送する形式に変換するイベント合成部と、

入力イベントを受け取り、入力制御部へ転送する入力インターフェース部と、イベント合成部から受け取ったイベントを接続先へ転送する出力インターフェース部とで構成される論理シミュレーション装置。

【請求項2】請求項1記載の論理回路シミュレーション装置を一つの処理単位として、複数の処理単位を、ネットワーク等の通信手段で相互に接続した並列処理方式の論理回路シミュレーション装置。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、論理回路シミュレーション装置、特に、イベント駆動式の論理シミュレーション装置に関する。

【0002】

【従来の技術】従来、論理回路のシミュレーションは、汎用機等でソフトウェア的に行われるものが主流であった。しかし、扱う回路規模の増大に汎用機の性能が追いつけなくなり、ソフトウェア処理に比較して4桁以上の高速化が必要となってきた。

【0003】このため、論理回路のシミュレーションを行うための専用ハードウェアが開発されるようになってきた。この専用ハードウェアによる論理シミュレーション装置は、複数の処理単位に負荷を分散し、演算の種類を標準的なゲートとフリップフロップだけに限定して、高速性を追及している。

【0004】

【発明が解決しようとする課題】この従来の専用ハードウェアによる論理シミュレーション装置は、演算の種類を標準的なもの、例えば、4入力1出力の標準ゲートだけに限定している。従って、扱える回路はゲートレベルのものに限定され、機能レベルの回路のシミュレーションは不可能であった。

【0005】また、機能レベルのシミュレーションをソフトウェアで実現したものは、シミュレーション速度が遅く、速やかな論理検証を行うことができなかった。

【0006】

【課題を解決するための手段】本発明の装置は、論理回路を、ゲートやメモリ等の機能を表すノードとノード間の接続情報とでモデル化し、ノードの出力変化をイベントとして次のノードに伝えて論理回路のシミュレーションを行う論理シミュレーション装置において、ノードの入力情報を記憶する入力メモリと、ノードの出力情報を記憶する出力メモリと、ノードの接続情報を記憶する接続メモリと、ノードと入力メモリのアドレス値の関係を記憶する索引メモリと、イベントが伝播したノードの情報を記憶するイベントメモリと、その他のノード情報等を記憶するデータメモリと、入力イベントを解析して検索部と更新部に指令を出す入力制御部と、イベントメモリを読み出してシミュレーションの実行指令を出す実行制御部と、発生したイベントの合成をイベント合成部に指令する出力制御部と、入力イベントを格納する入力メモリのアドレスを索引メモリの情報から決定する検索部と、入力イベントを入力メモリの検索部から渡されるアドレス位置に、イベント情報をイベントメモリに記録する更新部と、入力メモリからシミュレーション対象ノードの入力イベントを読み出し、演算部に与える入力ベクタを合成する入力合成部と、入力合成部から与えられる入力ベクタとデータメモリからのノード情報を用いてノードの演算を行う演算部と、演算部から与えられるノードの演算結果を、出力メモリに保持されている以前のノードの出力と比較して、イベント発生の有無を決定し、かつ出力メモリを更新するイベント発生部と、接続メモリからノードの接続情報を読み出し、イベント発生部から受け取ったイベントを接続先のノードへ転送する形式に変換するイベント合成部と、入力イベントを受け取り、入力制御部へ転送する入力インターフェース部と、イベント合成部から受け取ったイベントを接続先へ転送する出力インターフェース部とで構成される。

【0007】

〔実施例〕次に、本発明の実施例について図面を用いて説明する。

〔0008〕図1は本発明の一実施例を示す図である。

〔0009〕図1において、索引メモリ21、イベントメモリ22、入力メモリ23、出力メモリ24、接続メモリ25、データメモリ26は、それぞれ、ノードと入力メモリ23のアドレス値の関係、イベントが伝播したノードの情報、ノードの入力情報、ノードの出力情報、ノードの接続情報、その他のノード情報を記憶する。

〔0010〕検索部11は、入力イベントを格納する入力メモリ23のアドレスを決定し、更新部12は、イベントメモリ22と入力メモリ23の内容を更新する。入力合成部13は、演算部14に与える入力ベクタを合成し、演算部14は、ノードの演算を行う。イベント発生部15は、ノードの新出力値と旧出力値を比較し、イベント発生の有無を決定する。イベント合成部16は、ノードの出力値とノードノ接続先情報を合成して通信手段(図示省略)へ送る。

〔0011〕入力制御部2は、入力イベントを解析し、検索部11と更新部12を制御する。実行制御部3は、イベントメモリ22の情報を基にシミュレーションするノードを決定し、入力合成部13、演算部14、イベント発生部15を制御する。出力制御部4は、イベント合成部16を制御し、シミュレーションで発生したイベントの合成を行う。

〔0012〕入力インターフェース部1は、外部から入力イベントを受け取り、入力制御部2へ転送する。出力インターフェース部5は、イベント合成部16から受け取ったイベントを接続先へ転送する。入力端子31、出力端子32は、この装置をネットワーク等の通信手段に接続するためのものである。

〔0013〕この装置を単独で使用する場合は、入力インターフェース部1と、出力インターフェース部5の間に、イベントを転送するための接続が必要である。この図1では、この装置を図2の処理単位42として使用するので、接続を省いてある。

〔0014〕図2は本発明の他の実施例を示す図である。

〔0015〕図2において、ネットワーク41は、各処理単位間を相互に接続し、情報を転送する。処理単位42は、図1に示したシミュレーション装置全体をあらわす。マスタプロセッサ43は、各処理単位42へ指令を出し、同期をとる。また、ホストマシン44との通信を受け持つ。ホストマシン44は、論理回路モデルの作成、各処理単位へのモデルデータの転送、シミュレーション結果の収集および編集作業を受け持つ。

〔0016〕次に、論理回路のシミュレーション例を、図3を使用して説明する。図3に示す回路は、入力端子51に入力される不規則なパルスの立上りエッジを検出するためのものである。リセット入力端子52は、2つ

のフリップフロップ56と57をリセットするためのものである。入力端子53には、クロック信号を入力する。ゲート54は、フリップフロップ56への入力をつくり、ゲート55は、フリップフロップ57への入力をつくる。出力端子59は、検出した立上りエッジに対応したパルスを出力する端子であり、出力端子58は、フリップフロップ56の動作を監視するためのものである。

〔0017〕この回路をモデル化すると、回路の構成要素51～59がノード、その間の配線がネットに分解され、ノードの実現する機能、入力数、出力数、その他の情報はデータメモリ26に、ネット情報は接続メモリ25にそれぞれ格納される。また、ノードに割り振られた入力メモリ23のアドレスを索引メモリ21に格納する。

〔0018〕図4は、この回路に与える入力ボタンと、その出力ボタンである。61は入力端子53へのクロック、62は入力端子51へのパルス、63は入力端子52へのリセット信号の各入力ボタンである。また、64は出力端子58、65は出力端子59の出力ボタンである。ここでは、入力ボタンはマスタプロセッサ43から処理単位42にイベントとして送られ、出力ボタンは、逆に処理単位42からマスタプロセッサ43へイベントとして送られるものとする。また、a～hは、クロック61の立上りを示す。

〔0019〕入力ボタンが変化すると、マスタプロセッサ43から入力イベントが入力端子51～53へ送られる。イベントは、ネットワーク41を通して入力インターフェース部1へ入力され、入力制御部2へ転送される。

〔0020〕入力制御部2は、転送されたイベントを解釈して、検索部11と更新部12へ入力端子51への入力であることを伝え、イベントをセットするように指令を出し、次のイベント処理にかかる。以下、入力端子52、53へのイベントも同様に処理される。

〔0021〕検索部11は、索引メモリ21から入力端子51に対応する入力メモリ23のアドレスを検索し、更新部12に伝える。更新部12は、入力端子51へイベントがセットされたことをイベントメモリ22に書き込み、同時に検索部11からアドレス値を受け取る。次に、アドレスを入力メモリ23へセットし、そのアドレスの示す値をイベントの入力値に更新する。

〔0022〕実行制御部3は、マスタプロセッサ43からの同期信号によってシミュレーションを開始する。まず、イベントメモリ22からイベント情報を読み出し、入力端子51にイベントが入力されたことを知る。次に、入力合成部13と演算部14に、入力端子51のシミュレーションを指示する。また、イベント発生部15に、入力端子51のシミュレーション前の出力値の読み出しを指示する。次に、出力制御部4に入力端子51の

5

シミュレーションを行うことを伝える。全ての指示を終え、次のイベントの読み出しを行う。

【0023】入力合成部13は、実行制御部3からの指示をうけて、入力メモリ23から入力端子51の入力値を読み出し、それを演算部14に送る。

【0024】指示されたノードがゲート54のように複数の入力を必要とする場合は、入力メモリ23から全ての入力値を読み出し、それをベクタ形式に合成して演算部14に送る。

【0025】演算部14は、実行制御部3からの指示をうけて、データメモリ26から入力端子51のデータを読み出し、実行すべき演算を決定する。また、イベント発生部15に、出力メモリのアドレスを伝える。次に、入力合成部13から送られた入力ベクタを取り込み、演算を行い、演算結果をイベント発生部15に送る。

【0026】イベント発生部15は、実行制御部3からの指示をうけて、出力メモリ24から入力端子51のシミュレーション前の出力値を読み出す。この読み出しアドレスは、演算部14から指示される。次に、演算部14から送られた演算結果を取り込み、前の出力値と比較する。演算結果が出力値と異なる場合は、イベントが発生し、その値をイベント合成部16に転送し、併せて出力メモリ24の値を更新する。同時に、イベントの発生を出力制御部4にも伝える。仮に、演算結果が出力値と等しい場合は、イベントは発生しないので、値の転送は行わない。また、出力制御部4には、イベントが発生しなかったことを伝える。

【0027】ここで、対象ノードがフリップフロップ56のように複数の出力を持つ場合は、各々の出力を個別に判定し、変化した出力のみがイベントを発生する。変化しなかった出力にはイベントが発生しないので、不要なイベントの転送を避けられる。

【0028】出力制御部4は、実行制御部3からの連絡をうけて、入力端子51のイベントの合成準備をイベント合成部1に指示する。次に、イベント発生部15からの連絡により、イベントが発生したことを知り、イベントの合成開始をイベント合成部16に指示して、実行制御部3からの次の連絡を待つ。イベント合成部16から合成の終了が伝えられると、再び次のノードのイベント合成準備を指示する。イベントが発生しなかった場合は、合成の中止をイベント合成部16に指示し、実行制御部3からの次の連絡を待つ。

【0029】イベント合成部16は、合成準備の指示をうけて、接続メモリ25から入力端子51の接続先を読み出す。次に、合成開始の指示をうけて、イベント発生部15から受け取った値と接続先の情報を合成し、出力インターフェース部6に転送する。接続先が複数の場合はこれを繰り返し、全ての転送が終了した時点で、その旨を出力制御部4に伝える。

【0030】出力インターフェース部6は、イベント合

6

成部16から送られたイベントをネットワーク41に転送する。イベントは、ネットワーク41を通して接続先のノードをシミュレーションする処理単位42に転送される。

【0031】次に、機能レベルの論理シミュレーション例を説明する。

【0032】図5は、演算器のモデルである。命令レジスタ71は、クロック入力端子76からのクロック信号に同期して命令入力端子75から命令を取り込み、各ユニットに命令を伝える。レジスタファイル72は演算の対象となるデータを保持し、演算ユニット73へデータを与える。また、クロック信号に同期して、演算ユニット73の演算結果、もしくはデータ入力端子77からの入力データを取り込む。演算ユニット73は、命令レジスタ71からの命令をうけて演算を行う。出力バッファ74は、データ出力端子78へ演算結果を出力するか否かを制御する。

【0033】このような機能レベルの回路図であっても、各ユニット71～78をノードとして、その機能等のデータをデータメモリ26に、命令およびデータの転送ルートをネットとして、その接続関係を接続メモリ25に格納することで、図3の回路と同様にシミュレーションを行うことができる。

【0034】例えば、1回目のクロックで命令レジスタ71に加算命令がロードされたとする。次のクロックで、この加算命令はレジスタファイル72、演算ユニット73、出力バッファ74に送られる。ここの部分のシミュレーションは、先の説明の入力端子51の例と同様の手順で行われ、命令がイベントとして各ユニットをモデル化したノードに転送される。

【0035】各ノードがイベントを受け取ったことがイベントメモリ22に記録され、その値は入力メモリ23に記録される。実行制御部3が、イベントメモリ22からレジスタファイル72のノードのイベントを読み出し、シミュレーションを指示する。入力合成部13は、入力メモリ23からノードの入力を読み出し、演算部14に転送する。

【0036】演算部14は、ノードが表す機能をデータメモリ26から読み出し、入力合成部13からの入力（命令）に対して、出力にレジスタファイルのデータを送る演算を行う。このデータは、イベント発生部15からイベント合成部16に送られ、演算ユニット73へ向かうイベントが合成され、転送される。

【0037】以下、同様のシミュレーションを行い、最後にデータの加算結果がレジスタファイル72に転送され、入力端子76のクロックの変化で、この結果をレジスタファイル72に格納するシミュレーションを行う。

【0038】このように、論理回路がゲートレベルであろうと機能レベルであろうと、同様の手順でノードのシミュレーションを行う。各部は割り当てられた作業のみ

BEST AVAILABLE COPY

を行い、パイプライン処理で同時に作業を進める。この時、各部が必要とする情報は各々専用のメモリに割り当て、メモリ競合によるオーバーヘッドを最小限に抑える。

【0039】本発明では、シミュレーション中に起きるメモリ競合はイベントメモリ22と入力メモリ23に対するものだけであり、しかも、更新部12からのアクセスは書き込みのみ、実行制御部3と入力合成部13からのアクセスは読み出しのみである。従って、この2つにデュアルポートメモリを用いればメモリ競合は避けられる。

【0040】更に、このシミュレーション装置をネットワーク結合することで、各処理単位を並列に実行し、大規模な論理回路を極めて高速にシミュレーションできる。

【0041】

【発明の効果】以上説明したように、本発明は、シミュレーションを大きく入力・実行・出力の三つのフェーズに分割し、各フェーズを更に細分化して、全部で六つのユニットを同時に動かすことでノードのシミュレーションを流れ作業で行い、かつ、各ユニットが必要とする情報を個別の専用メモリに置くことで、メモリの競合を最小限に抑え、並列処理効率を高めることで、シミュレーションを高速に実行する。

【0042】更に、複数の装置をネットワークで接続することで、同時に多数のノードのシミュレーションを行い、大規模な論理回路のシミュレーションを高速に実行できる。

【0043】また、ノードの演算を機能レベルで表現でき、ノードの入出力に制限がないため、AND-OR等の複合ゲートや、レジスタファイル、加算器等を一つのノードとして扱え、回路を標準ゲートに展開する必要がない。

【図面の簡単な説明】

【図1】本発明の一実施例のブロック図である。

【図2】本発明の他の実施例のブロック図である。

【図3】ゲートレベルの論理回路の一例を示す図である。

【図4】回路の入出力チャートの一例を示す図である。

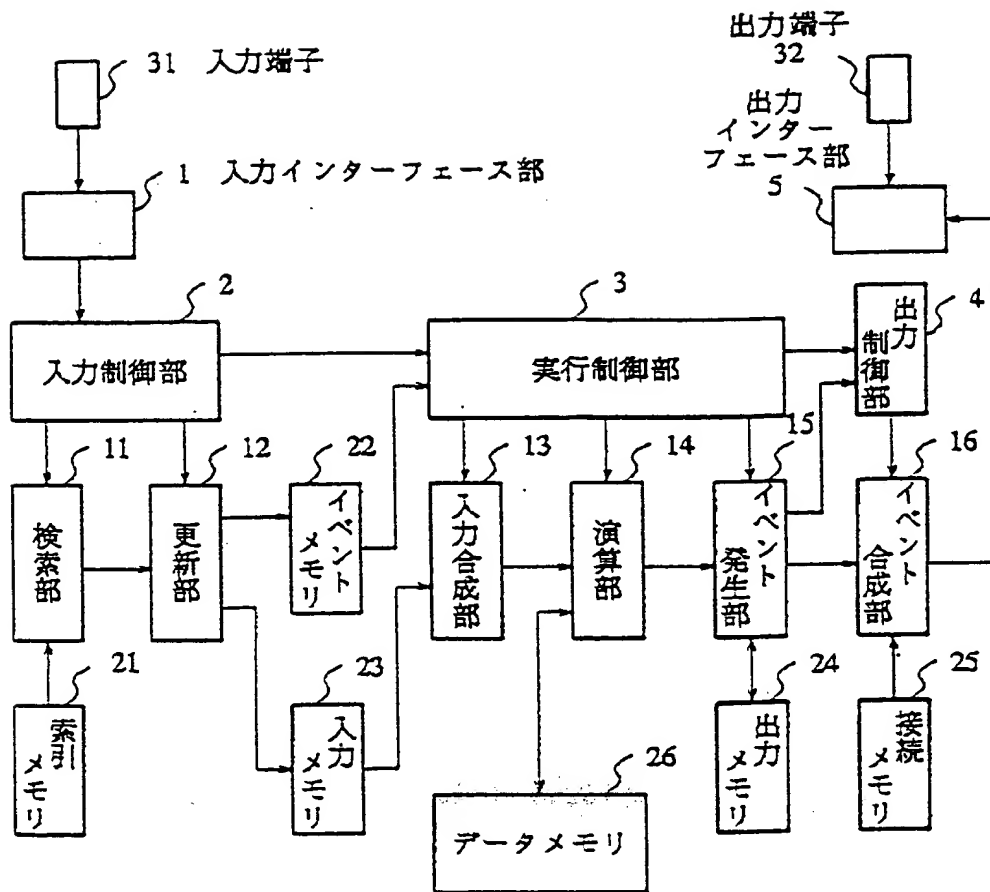
【図5】機能レベルの論理回路の一例を示す図である。

【符号の説明】

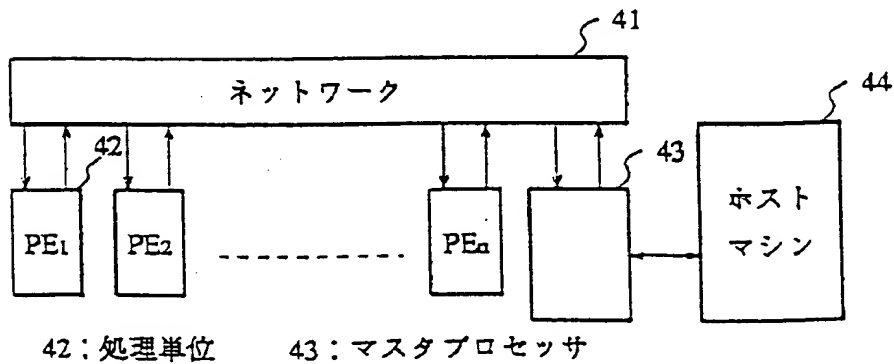
- 1 入力インターフェース部
- 2 入力制御部
- 3 実行制御部

- 4 出力制御部
- 5 出力インターフェース部
- 11 検索部
- 12 更新部
- 13 入力合成部
- 14 演算部
- 15 イベント発生部
- 16 イベント合成部
- 21 索引メモリ
- 22 イベントメモリ
- 23 入力メモリ
- 24 出力メモリ
- 25 接続メモリ
- 26 データメモリ
- 31 入力端子
- 32 出力端子
- 41 ネットワーク
- 42 処理単位
- 43 マスタプロセッサ
- 44 ホストマシン
- 51 入力端子
- 52 入力端子
- 53 入力端子
- 54 ゲート
- 55 ゲート
- 56 フリップフロップ
- 57 フリップフロップ
- 58 出力端子
- 59 出力端子
- 61 入力ボタン (CLK)
- 62 入力ボタン (IN)
- 63 入力ボタン (RST)
- 64 出力ボタン (QO)
- 65 出力ボタン (UP)
- 71 命令レジスタ
- 72 レジスタファイル
- 73 演算ユニット
- 74 出力バッファ
- 75 命令入力端子
- 76 クロック入力端子
- 77 データ入力端子
- 78 データ出力端子

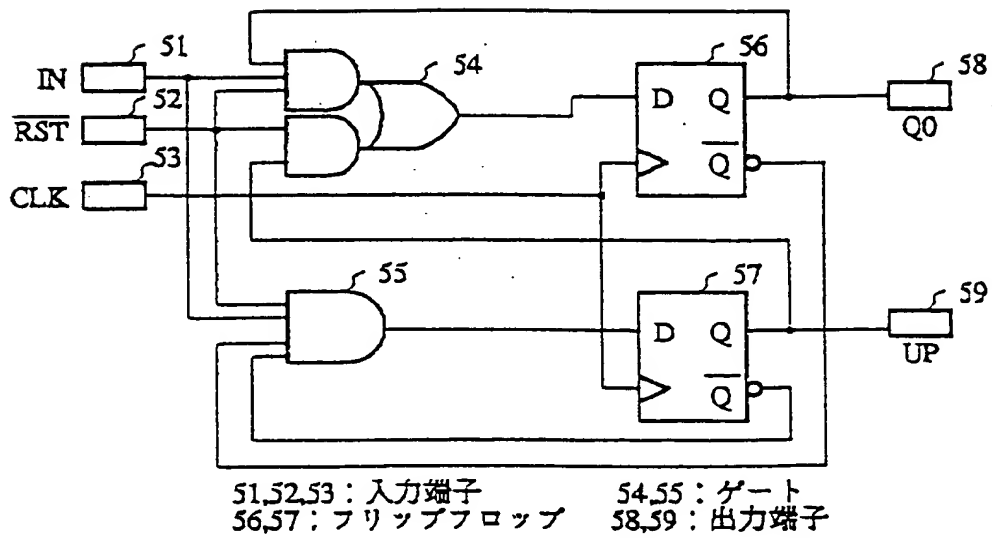
【図1】



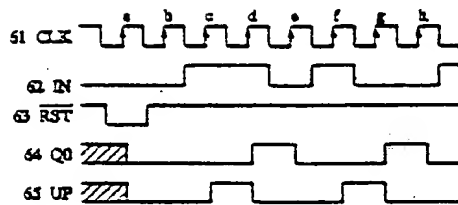
【図2】



【図3】



【図4】



【図5】

